

REMARKS

Claims 1-30 are pending in the present application. Independent Claims 1, 9, 15, 23, 29 and 30 were amended. Reconsideration of the claims is respectfully requested.

The Applicants thank the Examiner for discussing details of the Office Action in a telephone interview with the undersigned Attorney for the Applicants.

I. Amendments to the Specification

The Applicants have changed the description of Figure 6 to read Figures 6A and 6B. The change was made to conform the text to the drawings. The formal drawings split Figure 6 into two parts (6A and 6B) on two separate pages, whereas the informal drawings included Figure 6 on one page. No new matter has been added by this amendment.

II. 35 U.S.C. § 103, Obviousness

The Examiner has rejected Claims 1-30, under 35 U.S.C. § 103(a), as being unpatentable over U.S. Patent No. 5,511,196 to Shackelford et al. ("*Shackelford*"). This rejection is respectfully traversed.

As for Claim 1, the Office Action states:

As per claim 1, Shackelford teaches a method of updating an object association between a source object and a target object, comprising:

updating a target value holder of the target object to identify the source object and updating a source value holder of the source object to identify the target object; (col. 3 line 61 – col. 4 line 55) ("Thereafter, when a user modifies a data object, all associated reference objects are also modified to reflect the new state of the data object. Similarly, when a user modifies a reference object, the associated data object is also modified to reflect the new state of the reference object.") ("Although the content of reference objects 42, 44, and 46 is the same, each object has a unique identifying name.")

wherein a value holder is an object that wrappers a target or source object in order to implement a proxy pattern (col. 3 line 61 – col. 4 line 55) ("The relationship between reference object 42 and data object 40, and

between reference object 62 and data object 70 are strong relationships. Reference objects 42 and 62 will be maintained along with data objects 40 and 70. There will also be a validation that data objects 40 and 70 exist'). Shackelford does not specifically talk about transferring object associations over a network.

It is well know[sic] in the art of task management to transfer object associations over a network

It should be noted that what is claimed and what is disclosed in Shackelford differs only in the sense that in the claimed invention, object associations are being transferred over a network, while in Shackelford data reference objects are transmitted over a network. In either case, data is being transferred over a communications system, and the form or function of that data is immaterial to what the scope of invention is. Therefore, it would have been obvious to one of ordinary skill in the art to substitute the method of data reference objects as in Shackelford for object associations or any other type of data that is to be transmitted over a communications network. The discrepancy in type of data is immaterial, and the scopes of the inventions are essentially equivalent. One would want to transfer object associations over a network in order to maintain user information within different computer[sic] in an efficient and timely manner. This would reduce the cost of updating object associations within the network.

Office Action dated December 18, 2003, pages 2-4.

A prima facie case of obviousness is established when the teachings of the prior art itself suggest the claimed subject matter to a person of ordinary skill in the art. *In re Bell*, 991 F.2d 781, 783, 26 U.S.P.Q.2d 1529, 1531 (Fed. Cir. 1993). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994).

In other words, in comparing *Shackelford* to the claimed invention to determine obviousness, no limitations of the presently claimed invention may be ignored. As such, Claim 1 reads as follows:

1. A method of updating an object association between a source object and a target object, comprising:
updating a target value holder of the target object to identify the source object, the target value holder comprising a value holder that the target object points to;
and

updating a source value holder of the source object to identify the target object, the source value holder comprising a value holder that the source object points to; wherein a value holder is an object that wrappers a target object or source object in order to implement a proxy pattern.

Specifically, *Shackelford* does not disclose, teach or suggest the features of updating a target value holder of the target object to identify the source object, wherein the target value holder comprises a value holder that the target object points to, updating a source value holder of the source object to identify the target object, wherein the source value holder comprises a value holder that the source object points to, and a value holder that is an object that wrappers a target object or source object in order to implement a proxy pattern, as recited in Claim 1. Therefore, since *Shackelford* is missing these features of the claimed invention, Claim 1 is non-obvious and patentable over *Shackelford*.

More precisely, the Examiner points to the following sections of *Shackelford* as teaching the particular features of Claim 1:

With reference now to FIG 2., there is illustrated a block diagram of a data object 40, reference objects 44, and 46, and using objects 48 and 50 all within data processing system 8 in accordance with the method and system of the present invention. Reference objects 42, 44, and 46 and using objects 48 and 50 reside within an object oriented environment 52 while data object 40 resides within an application 54 outside of object oriented environment 52.

Multiple users are depicted by using object 48 and using object 50 within object oriented environment 52, which users may desire to access and modify data object 40, such as by deleting it or sending it to another application. A master reference object 44 is established and associated with data object 40. Copies of the master reference object 44 are then established and associated with each user. For example, reference object 42 is created and associated with using object 48 while reference object 46 is created and associated with using object 50. Therefore, each user within an object oriented environment has an associated reference object within the object oriented environment which then is associated with a data object in an application outside the object oriented environment. It should also be apparent that a reference object may be automatically created in response to a creation of a data object.

A reference object within an object oriented environment created in this manner may be associated with a data object external to the object oriented environment. Thereafter, when a user modifies a data object, all associated reference objects are also modified to reflect the new state of the data object. Similarly, when a user modifies a reference object, the associated data object is also modified to reflect the new state of the reference object.

...Referring to FIG. 3, there is depicted a block diagram of multiple users and associated reference objects established in accordance with the method and system of the present invention. As described above, a copy of master reference object 44 is created and associated with each user. In this manner, using object 48 is associated with reference object 42, using object 50 is associated with reference object 44, and using object 60 is associated with reference object 62. Although the content of reference objects 42, 44, and 46 is the same, each object has a unique identifying name. Therefore, there is a one-to-one relationship between reference objects and using objects, so that it is easy to maintain control of the reference objects. In this manner, it is clear which reference object is being utilized each time a using object creates, modifies, or deletes a reference object.

With reference to FIG. 4, there is illustrated a block diagram depicting strong and weak reference relationships created in accordance with the method and system of the present invention. As illustrated, a strong reference relationship exists between reference object 42 and data object 40. When a strong relationship exists, there will be a validation within object oriented environment 52 that a reference object has an existing data object, and that a data object has an associated reference object.

The relationship between reference object 42 and data object 40, and between reference object 62 and data object 70 are strong relationships. Reference objects 42 and 62 will be maintained along with data objects 40 and 70. There will also be a validation that data objects 40 and 70 exist.

Shackelford: Col. 3, lines 54-67; Col. 4, lines 1-55. As can be seen from these sections cited by the Examiner, *Shackelford* teaches that there is a one-to-one relationship between reference objects and using objects, that reference objects and using objects reside within an object oriented environment, while a data object resides within an application outside of the object oriented environment, and that the content of all of the reference objects is the

same, although each reference object has a unique identifying name. Therefore, according to *Shackelford*, each user within an object oriented environment has an associated reference object within the object oriented environment which then is associated with a data object in an application outside the object oriented environment.

Shackelford also teaches that when a user modifies a data object, all associated reference objects are also modified to reflect the new state of the data object. Similarly, when a user modifies a reference object, the associated data object is also modified to reflect the new state of the reference object. Additionally, *Shackelford* teaches that a strong reference relationship can exist between a reference object and a data object, and when such a strong reference relationship exists, there will be a validation within the object oriented environment that a reference object has an existing data object, and that a data object has an associated reference object.

As can be seen from these sections cited by the Examiner, *Shackelford* does not teach or suggest the claimed value holder, which is an object that wrappers a target object or source object in order to implement a proxy pattern. *Shackelford* only teaches that relationships exist between the using objects, reference objects and a data object. None of the using objects, reference objects, or data object disclosed in *Shackelford* is or suggests an object that wrappers a target object or source object in order to implement a proxy pattern, as recited in Claim 1.

Furthermore, assuming for the sake of argument, that a using object of *Shackelford* were to represent a source object as recited in Claim 1 (which it does not), and the data object of *Shackelford* were to represent a target object as recited in Claim 1 (which it does not), then following the same logic, a reference object of *Shackelford* would represent a value holder. However, even with this analogous (and incorrect) interpretation, *Shackelford* still would not teach or suggest the features of updating a target value holder of the target object to identify the source object, wherein the target value holder comprises a value holder that the target object points to, and updating a source value holder of the source object to identify the target object, wherein the source value holder comprises a value holder that the source object points to, as recited in Claim 1.

Moreover, assuming also for the sake of argument, that a master reference object of *Shackelford* were to represent a source object as recited in Claim 1 (which it does not), then using similar logic, the data object of *Shackelford* would represent a target object (which it does not). In this case, there would be no value holder (object) disclosed, taught or suggested in *Shackelford*. Specifically, as recited in Claim 1, a target value holder comprises a value holder that the target object points to, a source value holder comprises a value holder that the source object points to, and a value holder is an object that wrappers a target object or source object in order to implement a proxy pattern. In other words, Claim 1 recites at least four objects: (1) a source object; (2) a source value holder (object that the source object points to); (3) a target object; and (4) a target value holder (object that the target points to), wherein a value holder is an object that wrappers a target object or source object in order to implement a proxy pattern. As such, these objects or features and their unique relationships as recited in Claim 1 provide the one-to-one, one-to-many, many-to-one and many-to-many object associations of the present invention. *Shackelford* does not disclose, teach or suggest four such objects (e.g., source object, source value holder object, target object, target value holder object) and/or their unique relationships that provide the multiple object associations of the present invention.

As such, if the using object of *Shackelford* were to represent a source object of Claim 1 (which it does not), and the reference object of *Shackelford* were to represent a value holder of Claim 1 (which it does not), then since *Shackelford* teaches only a one-to-one relationship between reference objects and using objects, *Shackelford* does not teach or suggest a method that can solve the problems addressed by the present invention dealing with one-to-one, one-to-many, many-to-one, or many-to-many object associations, which are provided by the present invention. Therefore, for all of the above-described reasons, Claim 1 is not obvious over *Shackelford*.

The Examiner has also rejected independent Claims 9, 15, 23, 29 and 30 for essentially the same reasons given above with respect to the rejection of Claim 1. Independent Claims 9, 15, 23, 29 and 30 contain features similar to those of Claim 1 and are patentable over *Shackelford* for the same reasons given by the Applicants in response to the rejection of Claim 1. Furthermore, dependent Claims 2-8, 10-14, 16-22 and 24-28 depending from independent Claims 1, 9, 15, 23, 29 and 30 are also patentable for the

same reasons. Additionally, these dependent claims include other combinations of features not taught by *Shackelford*. Therefore, for all of the above-described reasons, the rejection of Claims 1-30, under 35 U.S.C. § 103(a), has been overcome.

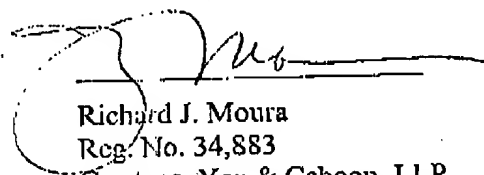
III. Conclusion

It is respectfully urged that the subject application is patentable over *Shackelford* and is now in condition for allowance.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: March 16, 2004

Respectfully submitted,



Richard J. Moura
Reg. No. 34,883
Carstens, Yee & Cahoon, LLP
P.O. Box 802334
Dallas, TX 75380
(972) 367-2001
Attorney for Applicants